How Level of Detail Expressions Work in Tableau

Version: 2021.3
Applies to: Tableau Desktop, Tableau Online, Tableau Public, Tableau Server

This article explains how level of detail expressions are computed and how they function in Tableau. For more information about LOD expressions and how they work, see the Understanding Level of Detail (LOD) Expressions whitepaper on the Tableau website.

Row Level Expressions and View Level Expressions

In Tableau, expressions referencing **unaggregated** datasource columns are computed for each row in the underlying table. In this case, the dimensionality of the expression is *row level*. An example of a row-level expression is:

```
[Sales] / [Profit]
```

This calculation will be evaluated in each row of the database. For each row, the Sales value in that row will be divided by the Profit value in that row, producing a new column with the result of the multiplication (a profit ratio).

If you create a calculation with this definition, save it with the name [**ProfitRatio**], and then drag it from the **Data** pane to a shelf, Tableau typically aggregates the calculated field for the view:

```
SUM([ProfitRatio])
```

By contrast, expressions referencing **aggregated** data source columns are computed at the dimensionality defined by the dimensions in the view. In this case, the dimensionality of the expression is view level. An example of a view-level expression is:

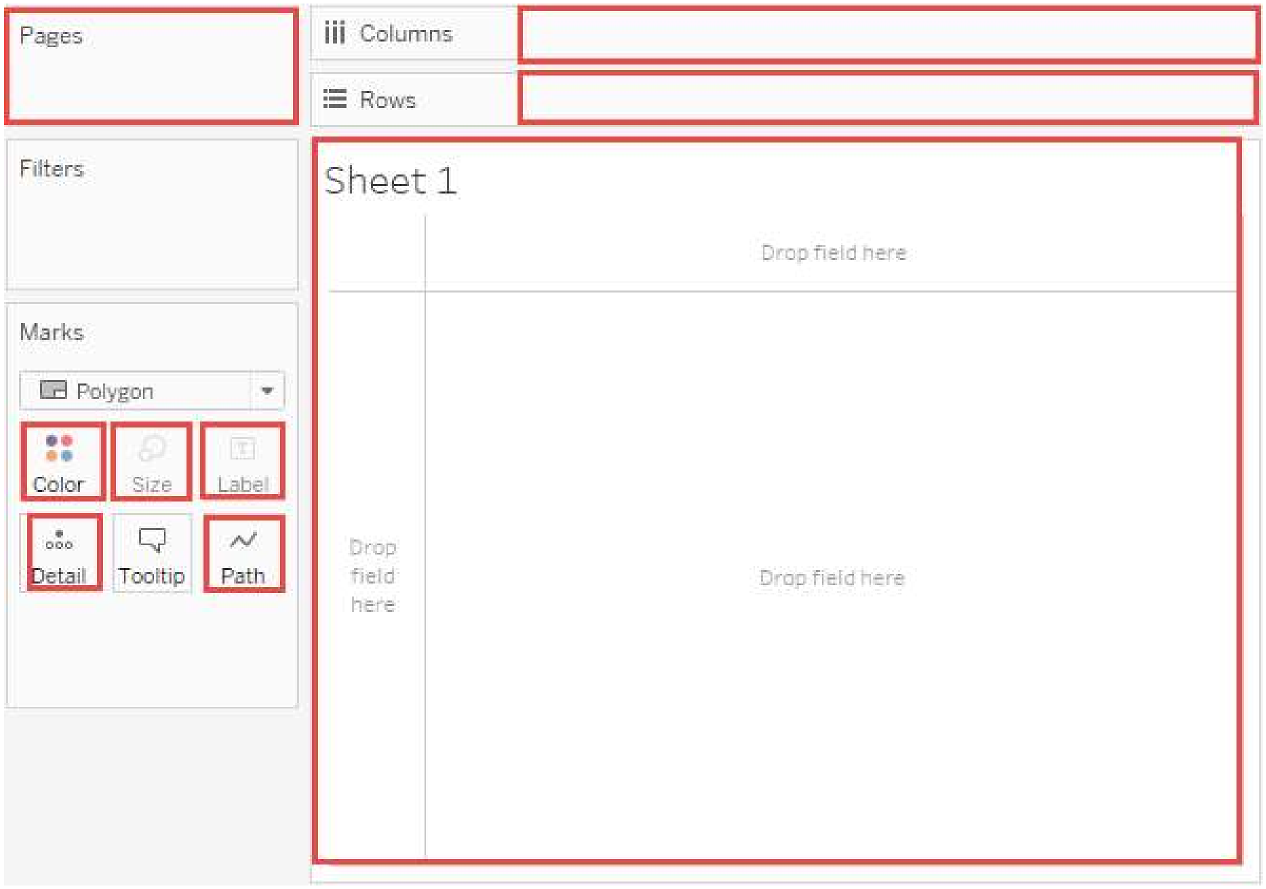
```
SUM(Sales) / SUM(Profit)
```

If you drag this calculation to a shelf (or type it directly on a shelf as an ad-hoc calculation), Tableau encloses it in an AGG function:

AGG(SUM(Sales) / SUM(Profit))

This is what is known as an aggregate calculation. For details, see Aggregate Functions in Tableau .

Dimension and set fields placed on any of the locations highlighted in the following image contribute to the view level of detail:



Before level of detail expressions were supported in Tableau, it was not possible to create calculations at a level of detail other than the view level. For example, if you attempt to save the following expression, Tableau displays the error message: “Cannot mix aggregate and non-aggregate arguments with this function”:

```
[Sales] - AVG([Sales])
```

The user’s intent in this case was to compare store sales for each individual store to the average of sales for all stores. This can now be accomplished with a level of detail expression:

```
[Sales] - {AVG([Sales])}
```

This is what is known as a table-scoped level of detail expression. See Table-Scoped (calculations_calculatedfields_lod.htm#Table)

Limitations for Level of Detail Expressions

The following limitations and constraints apply for level of detail expressions. Also see Data Source Constraints for Level of Detail Expressions.

- Level of detail expressions that reference floating-point measures can behave unreliably when used in a view that requires comparison of the values in the expression. For details, see Understanding data types in calculations .
- Level of detail expressions are not shown on the Data Source page. See Data Source Page (environment_datasource_page.htm#e).
- When referencing a parameter in a dimensionality declaration, always use the parameter name, and not the parameter value.
- With data blending, the linking field from the primary data source must be in the view before you can use a level of detail expression from the secondary data source. See Troubleshoot Data Blending (multipleconnections_troubleshooting.htm).

In addition, some data sources have complexity limits. Tableau will not disable calculations for these databases, but query errors are a possibility if calculations become too complex.

Level of Detail Expressions Can Be Dimensions or Measures

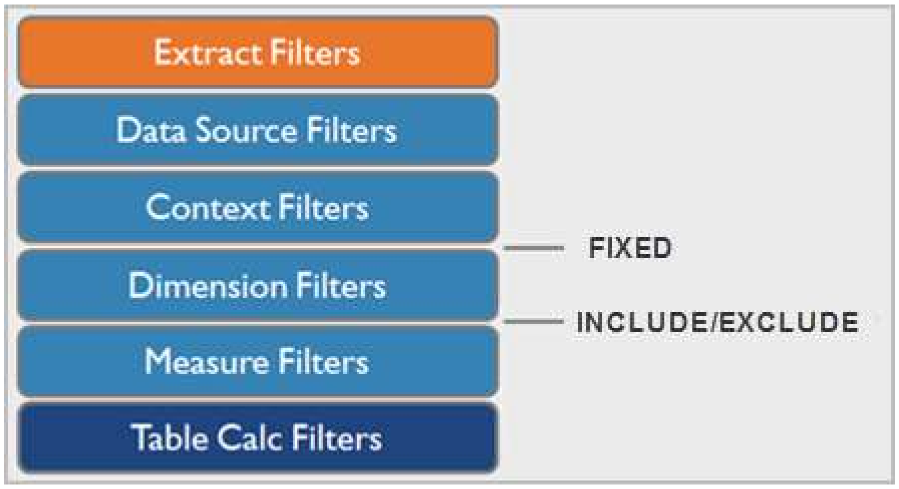
When you save a level of detail expression, Tableau adds it to either the Dimensions or the Measures area in the Data pane.

FIXED level of detail expressions can result in measures or dimensions, depending on the underlying field in the aggregate expression. So MIN([Date])} will be a dimension because [Date] is a dimension, and {fixed Store : SUM([Sales])} will be a measure because [Sales] is a measure. When a FIXED level of detail expression is saved as a measure you have the option of moving it to dimensions.

INCLUDE and EXCLUDE level of detail expressions are always measures.

Filters and Level of Detail Expressions

There are several different kinds of filters in Tableau and they get executed in the following order from top to bottom.



The text on the right shows where level of detail expressions are evaluated in this sequence.

Extract Filters (in orange) are only relevant if you're creating a Tableau Extract from a data source. Table calculations filters (dark blue) are applied after calculations are executed and therefore hide marks without filtering out the underlying data used in the calculations.

If you're familiar with SQL, you can think of measure filters as equivalent to the HAVING clause in a query, and dimension filters as equivalent to the WHERE clause.

FIXED calculations are applied before dimension filters, so unless you promote the fields on your Filter shelf to Improve View Performance with Context Filters (filtering_context.htm), they will be ignored. For example, consider if you have the following calculation on one shelf in a view, along with [**State**] on a different shelf:

```
SUM([Sales]) / ATTR({FIXED : SUM([Sales])})
```

This calculation will give you the ratio of a state's sales to total sales.

If you then put [**State**] on the Filters shelf to hide some of the states, the filter will affect only the numerator in the calculation. Since the denominator is a FIXED level of detail expression, it will still divide the sales for the states still in the view against the total sales for all states—including the ones that have been filtered out of the view.

INCLUDE and EXCLUDE level of detail expressions are considered after Dimension filters. So if you want filters to apply to your FIXED level of detail expression but don't want to use Context Filters, consider rewriting them as INCLUDE or EXCLUDE expressions.

Aggregation and Level of Detail Expressions

The level of detail of the view determines the number of marks in your view. When you add a level of detail expression to the view, Tableau must reconcile two levels of detail—the one in the view, and the one in your expression.

The behavior of a level of detail expression in the view varies depending on whether the expression's level of detail is coarser, finer, or the same as the level of detail in the view. What do we mean by "coarser" or "finer" in this case?

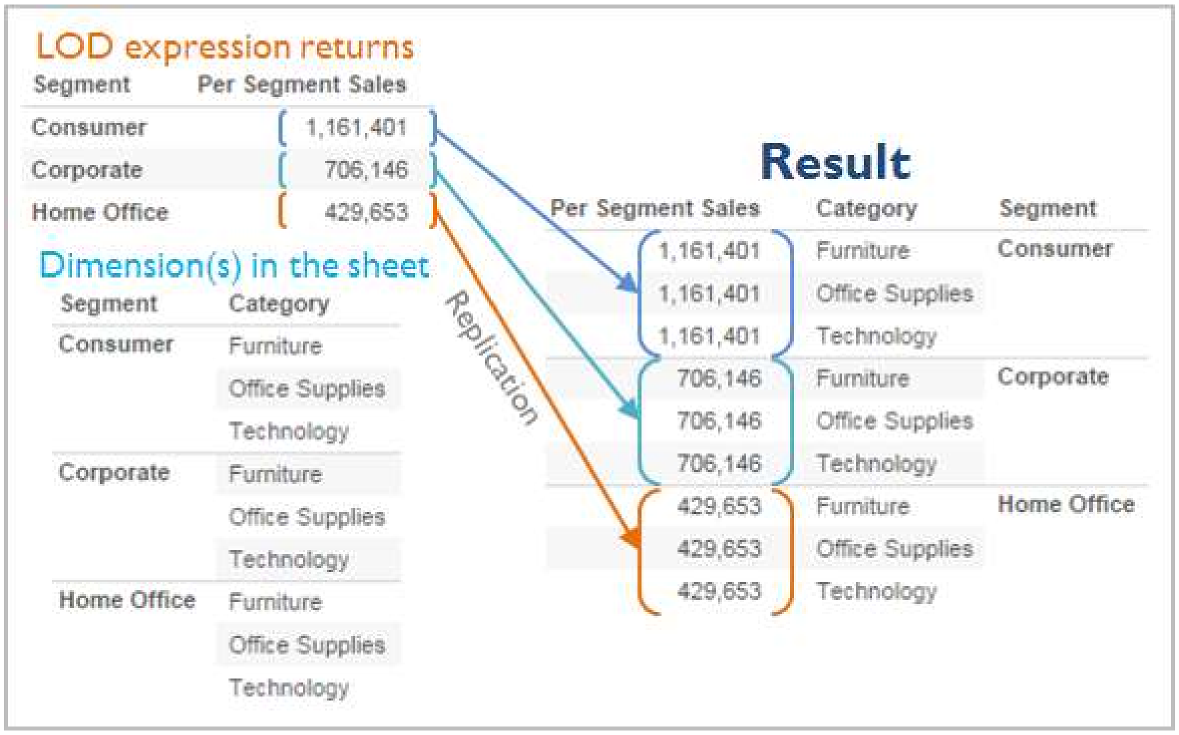
Level of Detail Expression is Coarser Than View Level of Detail

An expression has a coarser level of detail than the view when it references a subset of the dimensions in the view. For example, for a view that contained the dimensions [**Category**] and [**Segment**], you could create a level of detail expression that uses only one of these dimensions:

```
{FIXED [Segment] : SUM([Sales])}
```

In this case, the expression has a coarser level of detail than the view. It bases its values on one dimension ([**Segment**]), whereas the view is basing its view on two dimensions ([**Segment**] and [**Category**]).

The result is that using the level of detail expression in the view causes certain values to be replicated—that is, to appear multiple times.



Segment	Per Segment Sales
Consumer	1,161,401
Corporate	706,146
Home Office	429,653

Segment	Category
Consumer	Furniture
	Office Supplies
	Technology
Corporate	Furniture
	Office Supplies
	Technology
Home Office	Furniture
	Office Supplies
	Technology

Per Segment Sales	Category	Segment
1,161,401	Furniture	Consumer
1,161,401	Office Supplies	
1,161,401	Technology	
706,146	Furniture	Corporate
706,146	Office Supplies	
706,146	Technology	
429,653	Furniture	Home Office
429,653	Office Supplies	
429,653	Technology	

Replicated values are useful for comparing specific values against average values within a category. For example the following calculation subtracts average sales for a customer from the average sales overall:

```
[Sales] - {FIXED [Customer Name] : AVG([Sales])}
```

When values are being replicated, changing the aggregation for the relevant field in the view (for example, from AVG to SUM) will not change the result of the aggregation.

Level of Detail Expression is Finer Than View Level of Detail

An expression has a finer level of detail than the view when it references a superset of the dimensions in the view. When you use such an expression in the view, Tableau will aggregate results up to the view level. For example, the following level of detail expression references two dimensions:

```
{FIXED [Segment], [Category] : SUM([Sales])}
```

When this expression is used in a view that has only [Segment] as its level of detail, the values must be aggregated. Here's what you would see if you dragged that expression to a shelf:

```
AVG([{FIXED [Segment]], [Category]] : SUM([Sales]])}])
```

An aggregation—in this case, average—is automatically assigned by Tableau. You can change the aggregation as needed.

Adding a Level of Detail Expression to the View

Whether a level of detail expression is aggregated or replicated in the view is determined by the expression type (FIXED, INCLUDE, or EXCLUDE) and whether the expression's granularity is coarser or finer than the view's.

- INCLUDE level of detail expressions will have either the same level of detail as the view or a finer level of detail than the view. Therefore, values will never be replicated.

- FIXED level of detail expressions can have a finer level of detail than the view, a coarser level of detail, or the same level of detail. The need to aggregate the results of a FIXED level of detail depends on what dimensions are in the view.
- EXCLUDE level of detail expressions always cause replicated values to appear in the view. When calculations including EXCLUDE level of detail expressions are placed on a shelf, Tableau defaults to the ATTR aggregation (as opposed to SUM or AVG) to indicate that the expression is not actually being aggregated and that changing the aggregation will have no effect on the view.

Level of detail expressions are always automatically wrapped in an aggregate when they are added to a shelf in the view unless they're used as dimensions. So if you double-click on a shelf and type

```
{FIXED[Segment], [Category] : SUM([Sales])}
```

and then press Enter to commit the expression, what you now see on the shelf is

```
SUM({FIXED[Segment], [Category] : SUM([Sales])})
```

But if you double-click into the shelf to edit the expression, what you see in edit mode is the original expression.

If you wrap a level of detail expression in an aggregation when you create it, Tableau will use the aggregation you specified rather than assigning one when any calculation including that expression is placed on a shelf. When no aggregation is needed (because the expression's level of detail is coarser than the view's), the aggregation you specified is still shown when the expression is on a shelf, but it is ignored.

Data Source Constraints for Level of Detail Expressions

For some data sources, only more recent versions support level of detail expressions. Some data sources do not support level of detail expressions at all.

In addition, some data sources have complexity limits. Tableau will not disable calculations for these databases, but query errors are a possibility if calculations become too complex.

Data Source	**Support**
Actian Vectorwise	Not supported.
Amazon EMR Hadoop Hive	Supported for Hive 0.13 and later.
Amazon Redshift	Supported.
Aster Database	Supported for version 4.5 and later.
Cloudera Hadoop	Supported for Hive 0.13 and later.
Cloudera Impala	Supported for Impala 1.2.2 and later.
Cubes (multidimensional data sources)	Not supported.
DataStax Enterprise	Not supported.
EXASOL	Supported.
Firebird	Supported for version 2.0 and later.

Generic ODBC	Limited. Depends on the specific data source.
Google Big Query	Supported for standard SQL, not supported for legacy SQL.
Hortonworks Hadoop Hive	Supported for Hive 0.13 and later. On version 1.1 of HIVE level of detail expressions that produce cross joins are not reliable. Cross join occur when there is no explicit field to join on. For example, for a level of detail expression `{fixed [Product Type] : sum(sales)}` when the view only contains one dimension [**Ship Mode**], Tableau creates a cross-join. A cross join produces rows which combine each row from the first table with each row from the second table.
IBM BigInsights	Supported.
IBM DB2	Supported for version 8.1 and later.
MarkLogic	Supported for version 7.0 and later.
Microsoft Access	Not supported.

Microsoft Jet-based connections (legacy connectors for Microsoft Excel, Microsoft Access, and text)	Not supported.
Microsoft SQL Server	SQL Server 2005 and later.
MySQL	Supported.
IBM PDA (Netezza)	Supported version 7.0 and later.
Oracle	Supported version 9i and later.
Actian Matrix (ParAccel)	Supported version 3.1 and later.
Pivotal Greenplum	Supported for version 3.1 and later.
PostgreSQL	Supported version 7 and later.
Progress OpenEdge	Supported.
SAP HANA	Supported.
SAP Sybase ASE	Supported.
SAP Sybase IQ	Supported version 15.1 and later.
Spark SQL	Supported.
Splunk	Not supported.
Tableau Data Extract	Supported.
Teradata	Supported.
Vertica	Supported for version 6.1 and later.

See Also

Create Level of Detail Expressions in Tableau (calculations_calculatedfields_lod.htm)

Understanding Level of Detail (LOD) Expressions